Pile-up mitigation using attention

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
1 November 2021

REVISED
11 May 2022

ACCEPTED FOR PUBLICATION
19 May 2022

PUBLISHED
6 June 2022

PAPER

Pile-up mitigation using attention

B Maier[1,#,*], S M Narayanan[5,#,*], G de Castro[2], M Goncharov[3], Ch Paus[3] and M Schott[4]

[1] CERN, Geneva, Switzerland
[2] California Institute of Technology, Pasadena, CA, United States of America
[3] Massachusetts Institute of Technology, Cambridge, MA, United States of America
[4] Johannes-Gutenberg Universität Mainz, Mainz, Germany
[5] Currently at Flagship Pioneering
[#] These author contributed equally to this work.
[*] Authors to whom any correspondence should be addressed.

E-mail: benedikt.maier@cern.ch and sid.m.narayanan@gmail.com

Keywords: LHC, HL-LHC, pile-up, transformers, machine learning

Abstract

Particle production from secondary proton-proton collisions, commonly referred to as pile-up, impair the sensitivity of both new physics searches and precision measurements at large hadron collider (LHC) experiments. We propose a novel algorithm, Puma, for modeling pile-up with the help of deep neural networks based on sparse transformers. These attention mechanisms were developed for natural language processing but have become popular in other applications. In a realistic detector simulation, our method outperforms classical benchmark algorithms for pile-up mitigation in key observables. It provides a perspective for mitigating the effects of pile-up in the high luminosity era of the LHC, where up to 200 proton-proton collisions are expected to occur simultaneously.

1. Introduction

The large hadron collider (LHC) at CERN Geneva, will remain the most powerful tool on Earth to produce and study heavy elementary particles, at least for this decade. Further maximizing the potential of its experiments, such as ATLAS and CMS, and thereby increasing the chances of discovering new physics in the coming LHC runs, is of paramount importance. These runs will be characterized by an ever-increasing instantaneous luminosity, i.e. by a larger average number of simultaneous proton-proton collisions. During the final High Luminosity phase of the LHC (HL-LHC) starting in 2027, this number is expected to reach 200, which is almost an order of magnitude more than what was seen during Run 2 (2016–2018).

This poses an enormous challenge to the experiments, because their reconstruction algorithms have to identify interesting signatures among a large number of signals coming from secondary collisions. Reconstructed objects falsely attributed to the primary collision are called ‘pile-up’, and they can dramatically impair the sensitivity of an analysis. Especially in the Run 3 and High-Luminosity scenarios, removing pile-up contamination will become a primary objective. We present a method for identifying pile-up particles with the help of deep neural networks based on sparse transformers adapted from the field of natural language processing.

One example of a widely-used algorithm for rejecting pile-up particles is the charged-hadron subtraction as used in the CMS particle-flow (PF) algorithm [1]. It removes charged particles whose tracks have not been assigned to the primary vertex. As more sophisticated algorithms, PUPPI [2] and SoftKiller [3] aim at further identifying the pile-up component present among neutral particles.

First studies on the performance of pile-up mitigation using image recognition techniques [4] to identify pile-up contributions, and graph-based [5, 6] methods have focused on a subset of particles in the event clustered into hadronic jets or did not consider detector resolution effects. In addition, the weights calculated for each particle using the PUPPI algorithm are used as input features in these networks. Here, we present for the first time a machine learning algorithm relying only on raw reconstructed information that outperforms the classical benchmarks like PUPPI on event- and jet-level metrics, and in a realistic detector setting. This is a crucial step toward demonstrating the superiority of machine learning-based pile-up rejection on a global event level at future detectors and scenarios like the HL-LHC.

The paper is organized as follows. Firstly, a description of the setup and of the datasets is provided that we use for training and for performing the comparisons between the different pile-up mitigation algorithms. We then give a detailed explanation of the implementation of Puma. Third, we introduce the key metrics to benchmark the performance of the algorithms and provide details on the algorithm training. Finally, we quantify the performance of Puma.

2. Setup

2.1. The Delphes simulation framework

The goal of Delphes [7] is to allow the simulation of a multipurpose detector for phenomenological studies. The simulation includes a track propagation system embedded in a magnetic field, electromagnetic (ECAL) and hadron calorimeters (HCAL), and a muon identification system. Physics objects that can be used for data analysis are then reconstructed from the simulated detector response. These include tracks and calorimeter deposits and high level objects such as isolated electrons, jets, taus, and missing energy.

The Delphes framework allows for a fast simulation of an approximated detector response for typical LHC detectors using parameterized resolution and efficiency functions. The simulation includes a tracking system within a magnetic field, ECAL and HCAL as well as a muon systems. High-level objects like isolated electrons, particle jets or missing transverse energies are reconstructed using low level observables such as tracks and energy deposits in the calorimeters. The calorimeter systems within Delphes is finely segmented in η and ϕ and it is assumed that ECAL and HCAL have the same granularity, i.e. each ECAL cell has a corresponding cell in the HCAL. The geometrical center of each cell then defines the coordinate of the calorimeter energy deposit.

The stable charged particles on generator level with a minimal transverse momentum (e.g. $p_\mathrm{T} > 100\,\mathrm{MeV}$) undergo the track reconstruction. The track reconstruction efficiency as well as the resolution and scale is parameterized vs. p_T, η and ϕ.

Particles on generator level that reach the calorimeter system leave the fractions f_ECAL and f_HCAL in the electromagnetic and HCAL cells, respectively. The relevant cells can then be group together in one calorimeter tower. When several particles reach the same cells the total energy of one tower is simple the sum over all particles, that leave energies either in the ECAL or HCAL. Delphes assumes that all electrons and photons leave their full energy on the ECAL system, i.e. $f_\mathrm{ECAL} = 1.0$ and $f_\mathrm{HCAL} = 0.0$. Muons and neutrinos are assumed to leave no energy at all in the calorimeter system. All stable hadrons are treated with $f_\mathrm{ECAL} = 0.0$ and $f_\mathrm{HCAL} = 1.0$, with the exception of Kaons and Λ, where the values $f_\mathrm{ECAL} = 0.3$ and $f_\mathrm{HCAL} = 0.7$ are used.

The resolutions of the electromagnetic and the HCAL are independently parameterized in dependence of the particle kinematics, using a stochastic, a noise and a constant as parameters.

The CMS collaboration was one of the first that implemented a PF algorithm for the reconstruction and measurement of finale state objects, in order to maximize the use of sub-detector measurements for the event reconstruction. Since the full PF approach is rather complex, a simplified version is implemented within the Delphes framework, which results in two types of object collections: PF tracks and PF towers. Each tower is described by the total energy deposited in the electromagnetic and the HCAL, E_ECAL and E_HCAL, respectively. In addition, the total energy deposit originating from charged particles in a given tower is stored in the variables $E_\mathrm{ECAL,trk}$ and $E_\mathrm{HCAL,trk}$. It is important to note in this context, that Delphes assumes that the momentum of charged particles is always measured best by the tracking system. This allows to define the energy flow of a tower by

$$E^\mathrm{eflow}_\mathrm{Tower} = \max(0, \Delta_\mathrm{ECAL}) + \max(0, \Delta_\mathrm{HCAL}),$$

where $\Delta_\mathrm{ECAL} = E_\mathrm{ECAL} - E_\mathrm{ECAL,trk}$ and $\Delta_\mathrm{HCAL} = E_\mathrm{HCAL} - E_\mathrm{HCAL,trk}$. The PF algorithms then create a PF track for each reconstructed track and a PF tower with the energy $E^\mathrm{eflow}_\mathrm{Tower}$ if $E^\mathrm{eflow}_\mathrm{Tower} > 0$.

PF tracks describe therefore all charged particles with a good resolution. PF towers on the other hand, contain the energy information of neutral particles and charged particles that have not been reconstructed by the tracking system. In addition, also energy deposits due to the positive smearing of the calorimeters are taken into account. A detailed description of the algorithms can be found in [7], where also several validation studies are shown.

2.2. Collision datasets

To emulate the HL-LHC data taking situation, we generate Monte-Carlo simulations of LHC proton-proton collisions at $\sqrt{s} = 14\,\mathrm{TeV}$ using Pythia version 8.244 [8]. Pythia is used for both matrix-element generation and parton showering and hadronization, employing the parton shower tune 4 C [9], which also provides a description of effects from multi-parton interaction and the underlying event. The stable generator-level particles are subsequently passed through a model of the CMS detector built with Delphes version 3.4.3pre01, retaining the correspondence between reconstructed PF objects and the incident truth particles. The layout of the detector roughly corresponds to the Phase-II upgrade conditions at CMS, including novel, high-granularity forward detector components. The processes simulated are dileptonic $\mathrm{t\bar{t}}$ production, $\mathrm{Z}(\to \nu\bar{\nu})$+jets production, vector boson fusion (VBF) production of a Higgs boson with subsequent $\mathrm{H} \to \mathrm{c\bar{c}}$ or $\mathrm{H} \to$ dark matter decays, and soft-QCD production. For each of the studied SM processes, 200 thousand events have been generated for training and evaluation. On average, 140 soft-QCD events are mixed with the hard scattering event, sampled from a total of 50 million soft-QCD events. In the following, 'PF candidate' and 'particle' will be used interchangeably to refer to a PF object.

Very few simplifying assumptions are made in the simulation and reconstruction of the events: for stable, charged particles (electrons, muons, charged hadrons), the vertex assignment is assumed to be perfect. No underlying event is simulated. We note that apart from these assumptions, this study is based on a fast but sophisticated detector simulation that realistically models particle reconstruction efficiencies and smearing of track momenta and calorimeter tower energies.

2.3. Particle and event definitions

Each PF candidate is represented by a set of features: its Lorentz four-vector, the particle species, the electric charge, the corresponding vertex (if available), and local cluster information (see section 3.1 for the method to obtain event sub-clusters). An event is represented as a set of PF candidates, as well as the number of reconstructed vertices. For each particle, we also compute the target quantity to be learned:

$$y = \frac{E_{\mathrm{LV}}^{\mathrm{gen}}}{E_{\mathrm{LV}}^{\mathrm{gen}} + E_{\mathrm{PU}}^{\mathrm{gen}}}, \tag{1}$$

where $E_{\mathrm{LV}}^{\mathrm{gen}}$ is the summed energy of the incident generated particles from the leading vertex (LV), i.e. from the hard interaction, associated with the PF candidate. Similarly, $E_{\mathrm{PU}}^{\mathrm{gen}}$ is the summed energy of associated incident generated particles stemming from pile-up interactions. The quantity y is referred to as hard energy fraction in the following.

3. Modeling pile-up with sparse transformers

We formulate the problem of pile-up identification as one of regressing the hard energy fraction of each particle. That is, we define a model $g(\cdot;\Theta)$ to minimize the loss $\mathcal{L}(\cdot;\Theta)$:

$$\mathcal{L}\left(\{\mathbf{x}_i\},\{y_i\};\Theta\right) = \sum_j ||g(\{\mathbf{x}_i\};\Theta)_j - y_j||_2^2\,, \tag{2}$$

where $\mathbf{x}_i$ is the feature vector of particle i in a given event, y_i is the hard energy fraction of particle i, Θ denotes free parameters of the model g, and $g(\cdots)_j$ is the prediction of the model for particle j. Note that this prediction is conditioned on all particles in the event $\{\mathbf{x}_i\}$, but the loss is computed independently for each particle.

This energy fraction regression task is distinct from previous ML approaches to the pile-up problem [5, 6], which treat the problem as one of classification: a particle is uniquely identified as arising from a hard or pile-up vertex. This definition is no longer valid when considering the realistic scenario of a detector with finite spatial and energy resolutions. A measured PF candidate is frequently the result of several particles depositing energy in the same detector component. Accordingly, we train our models to decompose each detected particle into co-linear LV and PU components.

We parameterize g as a deep neural network PUMA: Pile-Up Mitigation using Attention. Attention was first introduced in 2014 [10] for neural machine translation of natural languages, and further developed as self-attending Transformer networks [11] for a multitude of natural language tasks.

The dynamics of particle decay and hadronization at the LHC share some conceptual similarities with natural language problems: much information of particle dynamics can be described in a local picture (in phase space), with some information requiring higher-order global abstraction (e.g. jet hadronization). This is analogous to a natural language sentence, where most words are closely coupled to nearby words, but some long-range dependencies do arise. Therefore, we hypothesize that a similar model parameterization may work in both scenarios.

Algorithm 1. Hierarchical particle clustering algorithm: w_{clust} is the maximum cluster size, and k is the number of clusters at each iteration. KMeans is a k-means clustering function that computes clusters on a cylindrical surface

```
Function IterativeCluster ($C$, $w_{\text{clust}}$, $k$) $\rightarrow$ clusters :
    If $|C| < w_{\text{clust}}$ then
        return $\{C\}$ ;
    else
        $\{C_1, \ldots, C_k\} \leftarrow$ KMeans$(k, C)$ ;
        $FC \leftarrow \{\}$ ;
        for $i = 1, \ldots, k$ do
            $FC \leftarrow FC \bigcup$ IterativeCluster$(C_i, w_{\text{clust}}, k)$ ;
        return $FC$ ;
```

At the core of Puma lie several transformer layers [11]. The full model can be written as:

$$\begin{aligned} h_{\text{embed.}} &= \text{MLP}_{\text{embed.}}(X) \\ h_{\text{enc.}} &= \text{Transformer}(h_{\text{embed.}}) \\ \hat{Y} &= \text{MLP}_{\text{dec.}}(h_{\text{enc.}}), \end{aligned} \tag{3}$$

where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ is the particle feature matrix, MLP are multi-layer perceptrons, and Transformer is a stack of Transformer layers.

In defining a model that uses full self-attention, a difficulty arises: the time and memory complexity of self-attention scales quadratically with the cardinality of the input set. In a typical particle collision with 140 pile-up interactions, the number of particles, N, can reach ten thousand. This makes training the model with reasonable batch sizes, even on large GPUs, prohibitively difficult.

To reduce the memory consumption of the transformer, we find a way to sparsify the self-attention mechanism. While many variants of sparse attention exist [12–14], we use the Longformer [15] implementation. Essentially, we construct a nearest-neighbors graph among the particles, such that the non-zero elements of the adjacency matrix are a subset of a banded-diagonal matrix. By limiting the size of this band, w, we ensure that the attention complexity scales as $\mathcal{O}(Nw) \ll \mathcal{O}(N^2)$.

3.1. Hierarchical particle clustering and sparsification

First, we cluster particles in a hierarchical fashion on the surface of a cylinder parameterized by $(\sin(\phi), \cos(\phi), \eta)$. The result of $\text{IterativeCluster}(\{\mathbf{x_i}\}, w_{\text{clust}}, k)$ (algorithm 1) is a set of clusters of particles, of size w_{clust} or smaller.

Once the clusters $\{C_1, \ldots, C_{n_c}\}$ have been computed, we define a unique ordering. For each cluster C_i, we compute three quantities:

$$p_{\text{T}}(C) = \sum_{j \in C} p_{\text{T}}^j, \quad \eta(C) = \sum_{j \in C} \frac{p_{\text{T}}^j \cdot \eta^j}{p_{\text{T}}(C)}, \quad \phi(C) = \sum_{j \in C} \frac{p_{\text{T}}^j \cdot \phi^j}{p_{\text{T}}(C)}. \tag{4}$$

These represent, respectively, the total transverse momentum of the cluster, the p_{T}-weighted pseudorapidity of the cluster, and the p_{T}-weighted azimuthal angle of the cluster. We define a permutation of the clusters π. It is initialized as:

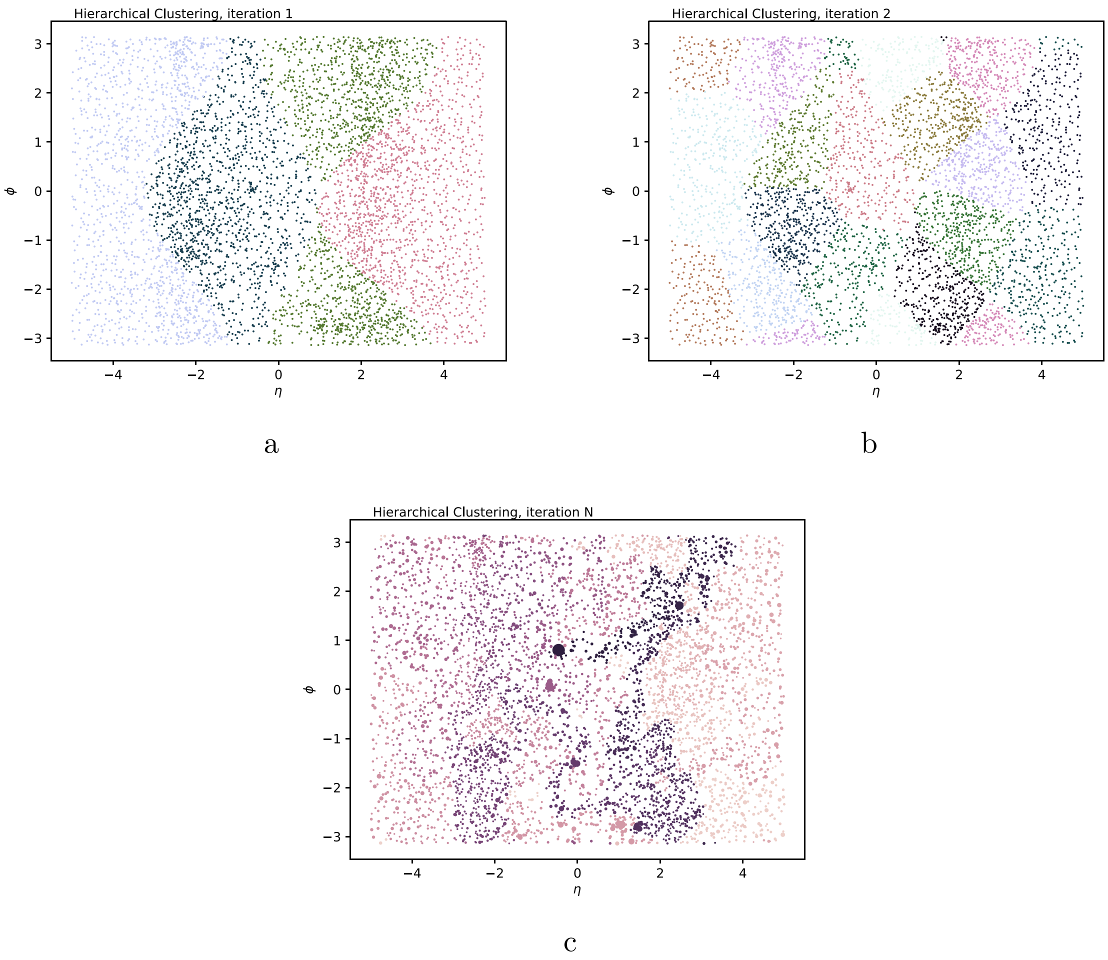
$$\pi(0) = \text{argmax}_i p_{\text{T}}(C_i). \tag{5}$$

Then, for $a > 0$:

$$\pi(a) = \text{argmin}_i \left\{ \Delta R(C_i, C_{\pi(a-1)}) \,\middle|\, i \neq \pi(b) \,\forall\, b < a \right\}, \tag{6}$$

where $\Delta R(x, y) = \sqrt{(\eta_x - \eta_y)^2 + (\phi_x - \phi_y)^2}$ is the L_2 metric in (η, ϕ) space. The results of this hierarchical clustering and ordering are shown for an example event in figure 1.

The ordering of the clusters give a natural ordering of the particles, in which the particles are first ordered according to the cluster they belong to, and within each cluster, by decreasing p_{T}. The particles can then be thought of as a graph with n_c complete, mutually-disconnected subgraphs. Each particle is a vertex, and two particles are connected by an edge if they belong two the same cluster. As a consequence of our particle ordering, the adjacency matrix A_{clust} is block-diagonal, as illustrated in figure 2. Furthermore, our cluster ordering means each block is adjacent to blocks arising from clusters that are proximal in (η, ϕ)-space. With



a

b

c

Figure 1. Particle assignment to clusters (indicated by color) for a certain event after one iteration, after two iterations, and after convergence. In all three figures, $k = 4$ and $w_{\mathrm{clust}} = 10$. In (c), clusters are colored based on their ordering (from dark to light), and particle marker areas are proportional to particle p_{T}.

$< n_{\mathrm{PU}} > = 140$, the number of reconstructed particles per event averages at about 6000, with some events having up to 9000 particles. Therefore, the sequence of particles is zero-padded up to a length of 9000 if fewer particles are reconstructed.

3.2. Banded attention

The clustering allows us to avoid attending over particles that are far away from the query particle. However, if we were to only allow the attention mechanism to consider keys adjacent to the query in A_{clust}, the model could not consider relationships between clusters. Instead, we use a banded-diagonal matrix A with bandwidth $w > w_{\mathrm{clust}}$. Larger values of w allow stronger cross-cluster attention within a Transformer layer. Figure 2 illustrates the relationship between $w_{\mathrm{clust}}, A_{\mathrm{clust}}$ and w, A. In practice, we find that values of w two to four times larger than w_{clust} allow for sufficient information flow between clusters. Attention band widths smaller than $w = 100$ allows us to leverage the efficient attention kernels provided by the authors of [15]. For widths larger than 100, we encounter memory limitations on our hardware. However, by stacking Transformer layers, we are effectively still able to increase the receptive field, as the transformed feature vectors after layer n carry information from particles $n \times w$ away from the query particle.

Puma is implemented in Python using PyTorch [16] and uses elements of the Transformers [17] library. The data processing, including IterativeCluster, is implemented in C++ using Delphes and ROOT [18].

4. Model training and evaluation

4.1. Training procedure

The standard MSE loss (equation (2)) puts all particles on equal footing. However, in a high pile-up event, we expect the vast majority of particles to be of extremely low-momentum, while the event dynamics are

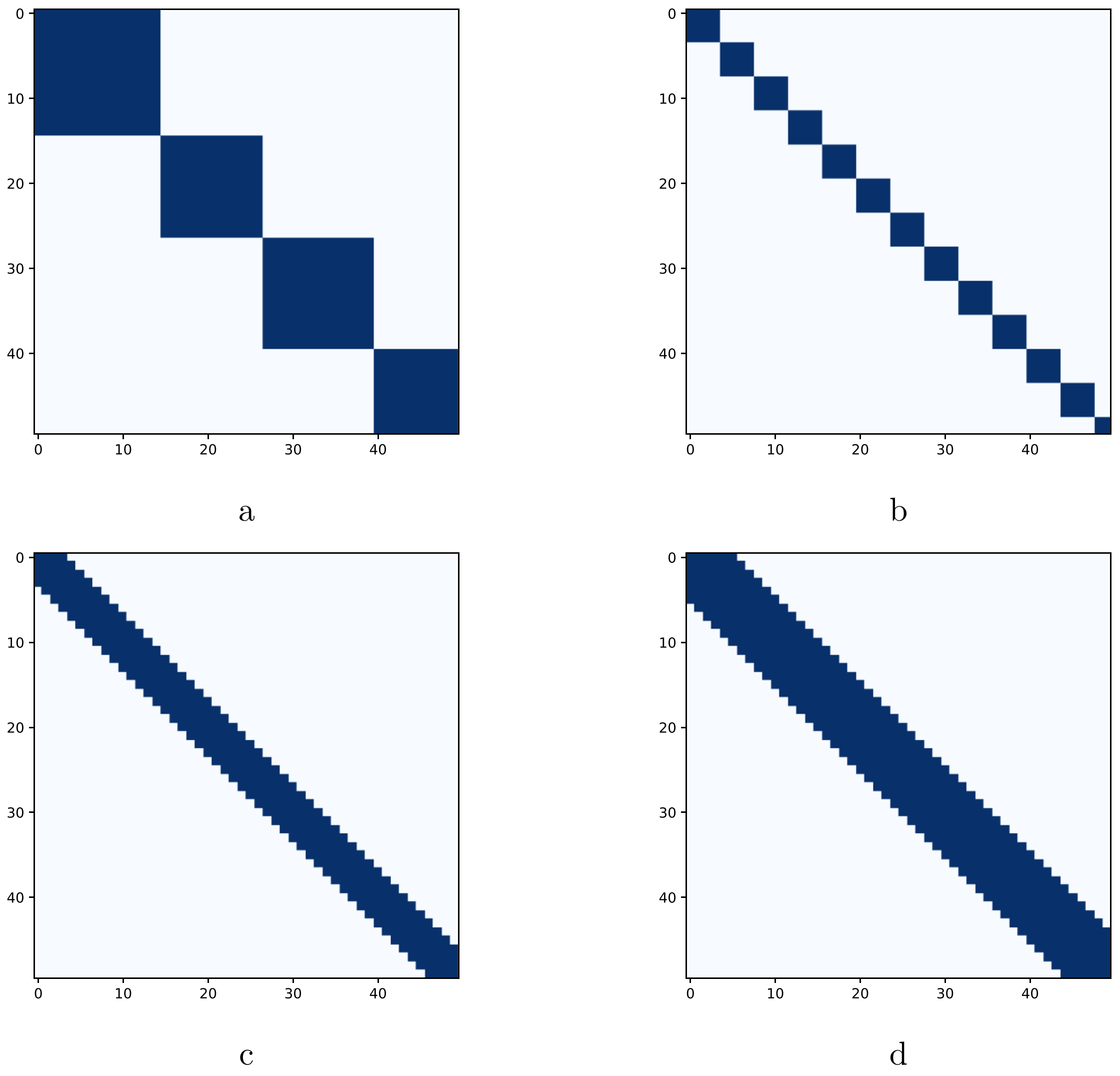


Figure 2. Figures (a) and (b): possible adjacency matrices for a 50-particle event, with $w_{\text{clust}} = 15$ and 5, respectively. Figure (c): the minimum banded-diagonal matrix required to cover the $w_{\text{clust}} = 5$ adjacency matrix. Figure (d): a slightly wider banded-diagonal matrix that allows for longer-range attention across clusters.

dominated by a handful of high-momentum particles. To bias PUMA toward more accurate modeling of high-momentum particles, we modify the loss:

$$\mathcal{L}\left(\{\mathbf{x}_i\},\{y_i\};\Theta\right) = \sum_j f_j || g(\{\mathbf{x}_i\};\Theta)_j - y_j||_2^2$$

$$f_j = \frac{p_{\mathrm{T},j}^{\alpha}}{\max_k p_{\mathrm{T},k}^{\alpha}}, \tag{7}$$

where k is the index over the particles contained in the leading cluster and $\alpha \geqslant 0$ is a tunable hyperparameter. We choose $\alpha = 2$ for our studies.

As vertex identification for charged particles is essentially perfect, PUMA only needs to optimize the loss function in equation (7) for neutral particles. This is equivalent to setting $f_j = 0$ for charged particles. In practice, we find doing so does not improve performance on neutral particle energy regression, nor other downstream metrics. In what follows, the loss is computed over all particles.

The loss function in equation (7) is minimized stochastically using the ADAM optimizer [19]. We employ batch sizes between 512 and 8192, distributed across four Nvidia Tesla V100 GPUs. To achieve the higher end of this range, we accumulate gradients over a maximum of eight iterations before applying weight updates. The learning rate is initialized to 10^{-3} and follows a cyclical schedule [20], with a period of four epochs and decay factor of 0.97. All models are trained to convergence, which typically occurs after $\mathcal{O}(10^5)$ steps.

4.2. Evaluation metrics

In addition to the key metric of the weighted mean square error, we evaluate the performance of pile-up mitigation techniques using three other metrics: the transverse momentum imbalance, ($p_\mathrm{T}^\mathrm{miss}$); the leading jet p_T; and the RMS of the transverse component of the hadronic recoil vector.

The vector sum of the p_T of all produced particles must be zero because the initial state of a hadron collision has no net momentum in the transverse plane. We calculate this as variable as:

$$\vec{p_\mathrm{T}}^\mathrm{miss} = -\sum_{i\in\mathrm{particles}} \vec{p_\mathrm{T}}_{,i}. \tag{8}$$

In events with undetectable particles (e.g. neutrinos), we expect to find $p_\mathrm{T}^\mathrm{miss} \equiv |\vec{p_\mathrm{T}}^\mathrm{miss}| > 0$. Many crucial Standard Model measurements, e.g. of the W mass, involve neutrinos and rely on an optimal pile-up rejection, and several beyond-SM models predict other undetectable particles (e.g. dark matter, stable SUSY particles, gravitons). Pile-up interactions are isotropically distributed in ϕ and have minimal $p_\mathrm{T}^\mathrm{miss}$ which causes an increase of the variance of reconstructed $p_\mathrm{T}^\mathrm{miss}$. Therefore, it is of great importance to assess the impact of pile-up, and pile-up mitigation techniques, on the missing transverse momentum. We define three per-event metrics from the momentum imbalance:

$$\hat{p}_x^\mathrm{miss} - p_x^\mathrm{miss}, \quad \hat{p}_y^\mathrm{miss} - p_y^\mathrm{miss}, \quad \hat{p}_\mathrm{T}^\mathrm{miss} - p_\mathrm{T}^\mathrm{miss}, \tag{9}$$

where p refers to the true momentum imbalance of generated particles prior to detector effects and $\hat{p}$ refers to the estimated momentum imbalance of reconstructed particles after detector effects, PF reconstruction, and pile-up mitigation.

As the LHC is a hadron collider, many events produce jets, i.e. bundles of collimated hadrons. In some cases, the jet is incidental to the process being studied (e.g. production of a Z boson at high p_T), while in other cases it is a fundamental signature of the process (e.g. VBF production of a Higgs boson). Pile-up interactions typically produce soft jets. However, even if the hard interaction has produced several hard jets, particles from the soft pile-up jets can affect the reconstruction of the hardest jets. Therefore, we also consider misreconstruction in the mass of the dijet system in the VBF Higgs production mode as another metric:

$$\hat{m}_{jj}^\mathrm{VBF} - m_{jj}^\mathrm{VBF}. \tag{10}$$

We do not expect any of these metrics to reach the ideal value of zero, even with perfect pile-up regression, because $\hat{p}$ includes the effects of detector resolution and PF reconstruction. Some previous work has neglected these effects, probably to isolate the impact of pile-up.

Finally, the measured hadronic recoil, $\vec{U}$, in a Z+jets event can be decomposed in a parallel, $U_{||}$, and a perpendicular, $U_\perp$, component with respect to the true vector boson transverse momentum, p_T^Z, where an ideal measurement yields $p_\mathrm{T}^\mathrm{Z} + U_{||} = 0$ and $U_\perp = 0$. The RMS error of the $U_\perp$ distribution is typically taken as a measure of the hadronic recoil resolution. However, this definition has the disadvantage that any rescaling of the measured hadronic recoil components by some factor α also changes the resolution.

$$U'_\perp = \alpha \cdot U_\perp, \qquad U'_{||} = \alpha \cdot U_{||}. \tag{11}$$

The width of the rescaled $U'_\perp$ distribution will be smaller for $\alpha < 1$, however no real gain in a physics measurement has been achieved, because an additional bias in $p_\mathrm{T}^\mathrm{Z} + U_{||}$ is introduced and the sensitivity of $U_{||}$ on p_T^Z decreases. In order to ensure a fair comparison between the different methods, the factors α_i in equation (11) for each bin of p_T^Z have been chosen such that the average bias $< p_\mathrm{T}^\mathrm{Z} + U_{||} >$ in a given bin of p_T^Z is the same for all methods. The figure of merit is then the RMS error of the resulting $U'_\perp$ distribution.

We define our gold standards as the event descriptions achievable assuming perfect pile-up regression but imperfect particle reconstruction. That is, we consider four scenarios:

- CHS: charged-hadron subtraction: only consider charged particles if they are associated with the primary vertex; leave neutral particles unscaled
- Puppi: scale each particle's 4-vector by the pile-up likelihood w_puppi
- PUMA: scale each particle's 4-vector by the estimated hard energy fraction $\hat{y}$
- Gold standard: compare to a sample generated with $n_\mathrm{PU} = 0$.

Table 1. Our choice of PUMA hyperparameters.

Parameter	Value
Embedding size	64
Hidden layer size	64
Number of attention heads	4
Attention band width	15
Number of transformers	12

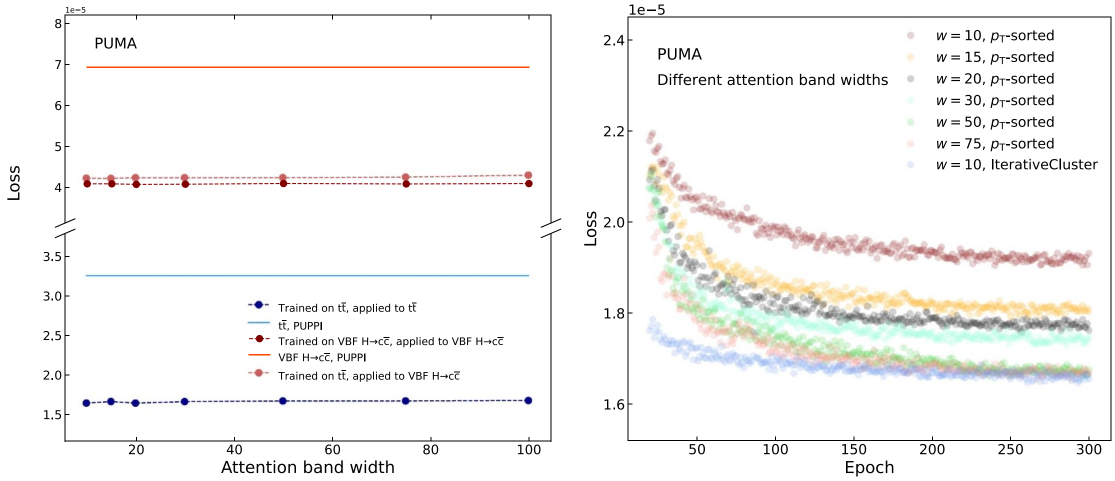


Figure 3. Left: PUMA pile-up identification performance is not strongly sensitive to the attention band width, having fixed the cluster size at $w_{\text{clust}} = 10$. The blue points show the average loss on $t\bar{t}$ events, using the default PUMA model trained on $t\bar{t}$ events. The light red points show the generalization of the default training to an entirely different process, VBF H($c\bar{c}$). Note that the difference in the absolute values of the two curves is not meaningful: the loss is dependent on a number of process-specific factors, like particle p_T, η, charge, etc. The purpose of this figure is to demonstrate that in both processes, the loss is not strongly a function of w. The dark red points show the optimal result for VBF H($c\bar{c}$), i.e. when PUMA is trained on this process instead of $t\bar{t}$. The increase in performance for this configuration is minimal considering the large improvement over PUPPI that is present for all PUMA scenarios. Right: if the PF candidate sequence is ordered by p_T, only very large attention band widths can recover the performance of ITERATIVECLUSTER with an attention band width of 10.

5. Results

5.1. Model hyperparameters

In this section we describe the final choice of model hyperparameters selected. The cluster size used in ITERATIVECLUSTER is set to $w_{\text{clust}} = 10$. The remainder of the hyperparameters are described in table 1. The model has 127 074 trainable parameters.

Of particular interest is the attention band width, as this is chosen to be small to limit the memory footprint of the model. As shown in figure 3, it turns out PUMA is robust across a range of attention widths. This holds true both for the process PUMA has been trained on, as well as for the inference for a different physics process, namely Higgs boson production with subsequent decays of the Higgs boson to charm quarks. We hypothesize this is due to the optimal cluster ordering, describing pile-up primarily as a local problem; additionally, by stacking many Transformer layers and thereby increasing the receptive field, we are sensitive to residual long-range relations between pile-up particles even with small attention band widths. As noted in section 3.2, we do not explore $w > 100$ due to computational constraints.

For inference, we consequently choose the model trained for 300 epochs with an attention band width of 15.

5.2. Physics performance

In terms of the key metrics introduced in section 4.2, we find that PUMA outperforms PUPPI across the board. Figures 4(a) and (b) show the difference between estimated p_T^{miss} and true p_T^{miss} for the same process that PUMA was trained on, i.e. dileptonic $t\bar{t}$ production. A sample that is statistically independent from the training sample has been used for inference. Note that the gold standard ($n_{\text{PU}} = 0$) does not have perfect reconstruction, due to finite detector and PF resolution. An absolute improvement in the p_T^{miss} resolution, computed as the variance of the respective distribution, of about 20% is observed compared to PUPPI.

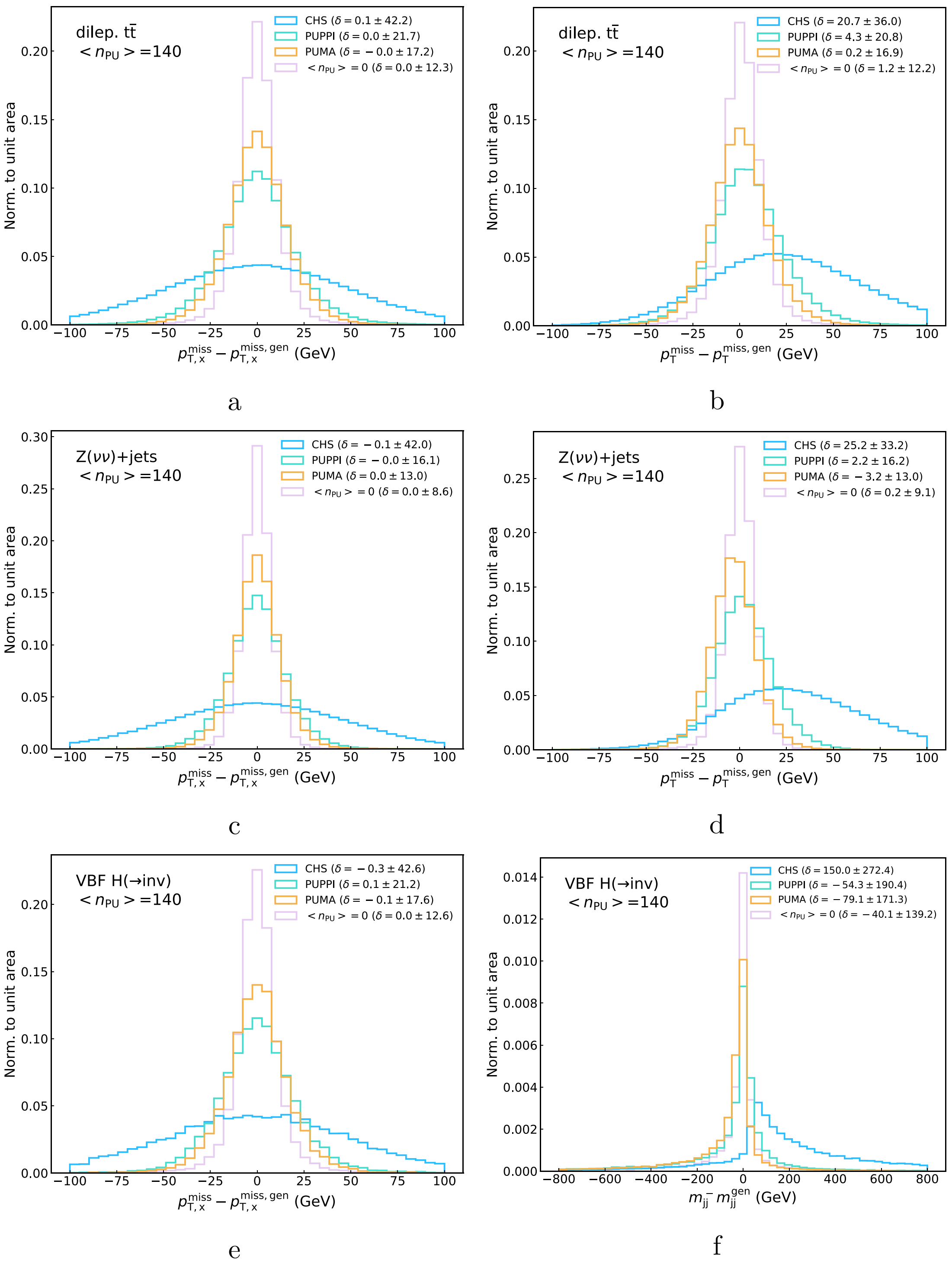


Figure 4. Resolutions in p_T^{miss} for top quark pair production with leptonic W boson decays in (a) and (b) are significantly improved for PUMA compared to other pile-up mitigation algorithms like CHS or PUPPI. Figures (c) and (d) show the resolution in p_T^{miss} in Z($\nu\nu$)+jets events, i.e. for a different physics process from what PUMA was trained on. Figures (e) and (f) show the resolution in p_T^{miss} and, respectively, dijet mass in VBF Higgs production events, with the Higgs boson decaying to invisible dark matter particles. The parameter δ is estimated for each distribution and consists of the mean and the variance, reflecting the resolution in the estimation of the respective quantity.

Alternatively, this can be characterized as PUMA bridging half of the gap between PUPPI and the theoretical minimum gold standard.

This demonstrates for the first time that a machine learning algorithm can mitigate pile-up effects more effectively than the currently best rule-based algorithm (PUPPI) at the event level, with a realistic detector simulation, and without actually using the scores of traditional algorithms like PUPPI as input features.

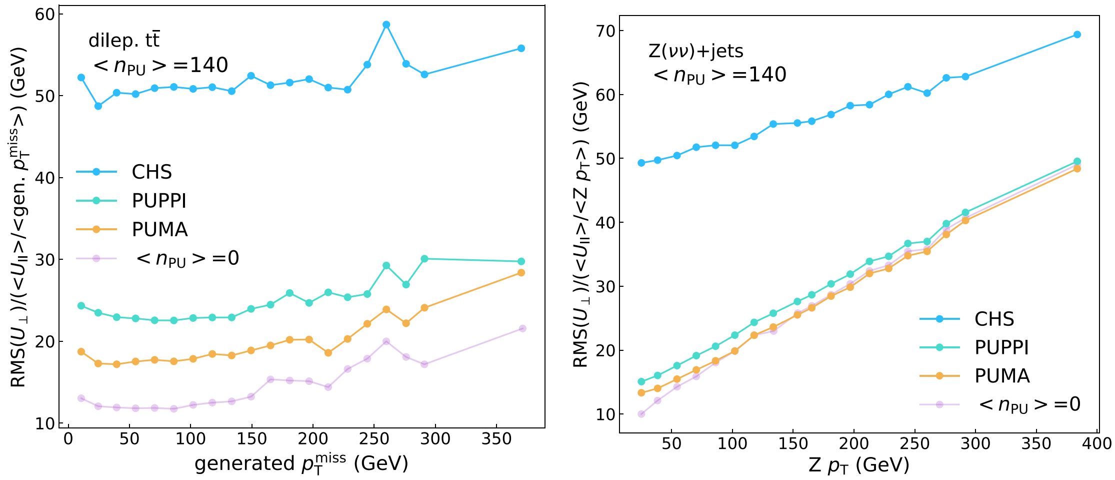


Figure 5. The RMS of the perpendicular hadronic recoil component for dileptonic top quark pair production and Z+jets production, as a function of $p_T^{\text{miss,gen}}$ and, respectively, the generated Z boson p_T. The recoil is scaled to have unity response.

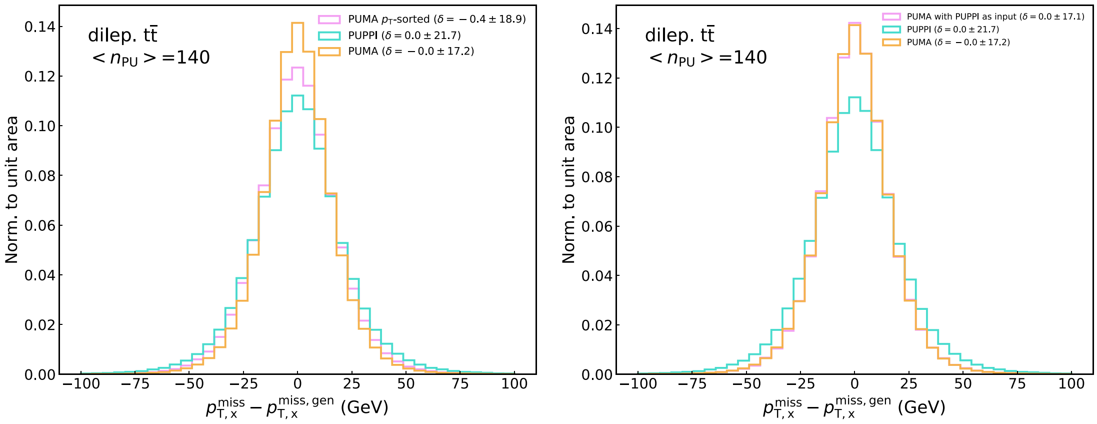


Figure 6. Left: comparison between two PUMA algorithms and PUPPI. An attention band width of 15 was used for both PUMA models; the PF candidate sequence is either ordered by p_T or according to ITERATIVECLUSTER, which gives the best performance. Right: using the PUPPI weight per particle as input in addition to the variables listed in table 2 does not result in sizable improvement over the default PUMA. In both figures, the parameter δ is estimated for each distribution and consists of the mean and the variance, reflecting the resolution in the estimation of the respective quantity.

The same improvement is achieved for a different physics process in figures 4(c) and (d), where the model was applied on a sample of Z($\nu\nu$)+jets events. Finally, as can be seen from figures 4(e) and (f), the dijet mass and p_T^{miss} for VBF Higgs production events with invisible decays of the Higgs boson—an essential search channel when looking for dark matter at the LHC—likewise demonstrate an increase in resolution compared to PUPPI.

The RMS error of the response-corrected $U_\perp$ distributions in $t\bar{t}$ production and Z+jets production is shown in figure 5, where a clear improvement over PUPPI is evident across the entire range of hadronic recoil. By eliminating virtually all contributions from pile-up in the case of Z+jets, we obtain the same resolution as observed in the gold standard sample.

Finally, figure 6 demonstrates that the particle grouping found by ITERATIVECLUSTER is necessary to achieve the observed performance. We compare to a version of PUMA trained using p_T-ordered particles. While momentum ordering does still lead to an improvement over PUPPI, this improvement is only half of what is observed with our optimal PUMA. This demonstrates that pile-up is to first order a local problem, where the particles in the vicinity of the query particle contain the most information about its vertex of origin. It also shows that attention mechanisms combined with ITERATIVECLUSTER are optimally suited to exploit this information. The marginal improvement shown in the right panel of figure 6 when using the PUPPI weight per particle as input feature in addition to the ones listed in table 2 suggests that PUMA manages to capture the information contained in PUPPI and further improves on it.

Table 2. Variables used as input to the network. While they are attributes of each particle, they resemble three categories of different locality: variables corresponding to properties of the individual particle (p_T, η, ϕ, E, particle ID, vertex ID); variables characterizing the cluster the particle is in (cluster ID, cluster R, cluster p_T); an event-wide variable (N_{PV}).

p_T	particle transverse momentum
η	particle pseudorapidity
ϕ	particle azimuthal angle
E	particle energy
particle ID	particle ID
vertex ID	vertex of particle, -1 if neutral
cluster ID	index of cluster containing particle
cluster ΔR	max. pairwise distance $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$ between particles found in cluster
cluster p_T^{ch}	scalar sum of p_T's of all charged LV particles in cluster containing the particle
cluster p_T^{neut}	scalar sum of p_T's of all neutral particles in cluster containing the particle
N_{PV}	number of reconstructed vertices in the event

The observed improvement in the resolutions in p_T^{miss} for a variety of processes, which are crucial for SM precision measurements and essential backgrounds in many searches for BSM physics, would directly translate to a more powerful analysis of a plethora of signatures expected in LHC collisions, especially toward the HL-LHC.

6. Summary

We have presented a highly effective pile-up mitigation algorithm Puma based on sparse self-attention. This method is the first machine learning approach relying only on raw reconstructed observables to demonstrate superior performance in a realistic detector scenario over current state-of-the-art, rule-based pile-up mitigation techniques. This holds true for both event-level and particle-level quantities. As pile-up effects will continue to worsen as the LHC moves to increasing luminosity, this is an important step toward showing that statistically-learned algorithms like sparse transformers can be very useful at the HL-LHC and beyond.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

The authors would like to thank Philip Harris and Lindsey Gray for productive conversations and feedback. The networks presented in this paper have been trained on the MIT-IBM Satori GPU cluster and on the Tier-2 MIT computing cluster.

This material is based upon work supported by the U.S. National Science Foundation under Award Number PHY-1624356 and the U.S. Department of Energy Office of Science Office of Nuclear Physics under Award Number DE-SC0011939.

Disclaimer

'This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.'

ORCID iDs

B Maier https://orcid.org/0000-0001-5270-7540
M Schott https://orcid.org/0000-0002-4235-7265

References

[1] Sirunyan A *et al* 2017 *J. Instrum.* **12** 10003

[2] Bertolini D, Harris P, Low M and Tran N 2014 *J. High Energy Phys.* **2014** 59

[3] Cacciari M, Salam G P and Soyez G 2015 *Europhys. J.* C **75** 59

[4] Komiske P T, Metodiev E M, Nachman B and Schwartz M D 2017 *J. High Energy Phys.* **2017** 51

[5] Martínez J, Cerri O, Spiropulu M, Vlimant J and Pierini M 2019 *Europhys. J. Plus* **134** 333

[6] Mikuni V and Canelli F 2020 *Europhys. J. Plus* **135** 463

[7] de Favereau J, Delaere C, Demin P, Giammanco A, Lemaitre V, Mertens A and Selvaggi M 2014 *J. High Energy Phys.* **2014** 57

[8] Sjöstrand T, Ask S, Christiansen J R, Corke R, Desai N, Ilten P, Mrenna S, Prestel S, Rasmussen C O and Skands P Z 2015 *Comput. Phys. Commun.* **191** 159–77

[9] Corke R and Sjostrand T 2011 *J. High Energy Phys.* **2011** 032

[10] Bahdanau D, Cho K and Bengio Y 2015 Neural machine translation by jointly learning to align and translate *Proc. ICLR* pp 1–15

[11] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L u and Polosukhin I 2017 Attention is all you need *Advances in Neural Information Processing Systems 30* ed I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan and R Garnett (Curran Associates, Inc.) pp 5998–6008

[12] Zhao G, Lin J, Zhang Z, Ren X and Sun X 2020 Sparse transformer: concentrated attention through explicit selection (available at: https://openreview.net/forum?id=Hye87grYDH)

[13] Child R, Gray S, Radford A and Sutskever I 2019 (arXiv:1904.10509)

[14] Malaviya C, Ferreira P and Martins A F T 2018 Sparse and constrained attention for neural machine translation *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Melbourne: Association for Computational Linguistics) pp 370–6

[15] Beltagy I, Peters M E and Cohan A 2020 Longformer: the long-document transformer (arXiv:2004.05150)

[16] Paszke A *et al* 2019 Pytorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems 32* ed H Wallach, H Larochelle, A Beygelzimer, B F d'Alché, E Fox and R Garnett (Curran Associates, Inc.) pp 8024–35

[17] Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M and Brew J 2019 (arXiv:1910.03771)

[18] Brun R and Rademakers F 1997 *Nucl. Instrum. Methods Phys. Res.* A **389** 81–86

[19] Kingma D and Ba J 2014 *Int. Conf. on Learning Representations*

[20] Smith L N 2017 Cyclical learning rates for training neural networks *2017 IEEE Winter Conf. on Applications of Computer Vision (WACV)* pp 464–72